PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06K 9/34

A2

(11) International Publication Number: WO 96/30860

(43) International Publication Date: 3 October 1996 (03.10.96)

(21) International Application Number: PCT/US96/04036

(22) International Filing Date: 21 March 1996 (21.03.96)

(30) Priority Data:
08/409,697 24 March 1995 (24.03.95) US

(71) Applicant: UNITED PARCEL SERVICE OF AMERICA, INC. [US/US]; 55 Glenlake Parkway N.E., Atlanta, GA 30328 (US).

(72) Inventor: HOLEVA, Lee, F.; 4485 Sheffield Court, Gurnee, IL 60031 (US).

(74) Agents: ANDERSON, Albert, S. et al.; Jones & Askew, 37th floor, 191 Peachtree Street, N.E., Atlanta, GA 30303-1769 (US).

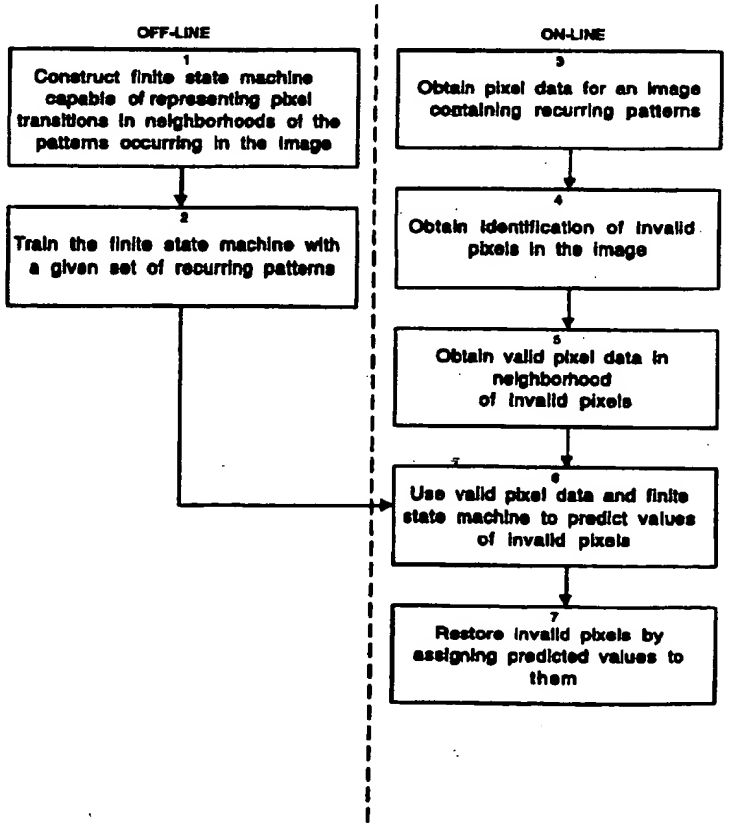
(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
*Without international search report and to be republished upon receipt of that report.*

(54) Title: METHOD AND APPARATUS FOR REMOVING SUPERIMPOSED LINES AND RESTORING PIXEL VALUES IN IMAGES CONTAINING RECURRING PATTERNS

(57) Abstract

The invention is a method and apparatus to restore the obscured poriton of an image of a sequence of characters or other patterns in the presence of superimposed lines. The character restoration is accomplished by means of two kinds of statistical information, the frequency of occurrence of image pixel value observations on either side of the obscuring line and the frequency of occurrence of character slice transitions within the width of the obscuring line. This statistical information is organized as a Hidden Markov Model. Having estimated the model's probabilities, the Viterbi algorithm is then employed to determine the optimal restoration across the length of the line. Two tasks are performed prior to the character restoration. First the Hidden Markov Model is trained. This task involves scanning across sequences of characters to estimate both observation and tansition probabilities. Model construction is done off-line and typically only once. Prior to the run-time character restoration, the line itself is located using a modified version of the Hough transform. Local eigenfits are employed to reduce the necessary computation.



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | |
|---|---|---|---|---|---|
| **AM** | Armenia | **GB** | United Kingdom | **MW** | Malawi |
| **AT** | Austria | **GE** | Georgia | **MX** | Mexico |
| **AU** | Australia | **GN** | Guinea | **NE** | Niger |
| **BB** | Barbados | **GR** | Greece | **NL** | Netherlands |
| **BE** | Belgium | **HU** | Hungary | **NO** | Norway |
| **BF** | Burkina Faso | **IE** | Ireland | **NZ** | New Zealand |
| **BG** | Bulgaria | **IT** | Italy | **PL** | Poland |
| **BJ** | Benin | **JP** | Japan | **PT** | Portugal |
| **BR** | Brazil | **KE** | Kenya | **RO** | Romania |
| **BY** | Belarus | **KG** | Kyrgystan | **RU** | Russian Federation |
| **CA** | Canada | **KP** | Democratic People's Republic of Korea | **SD** | Sudan |
| **CF** | Central African Republic | | | **SE** | Sweden |
| **CG** | Congo | **KR** | Republic of Korea | **SG** | Singapore |
| **CH** | Switzerland | **KZ** | Kazakhstan | **SI** | Slovenia |
| **CI** | Côte d'Ivoire | **LI** | Liechtenstein | **SK** | Slovakia |
| **CM** | Cameroon | **LK** | Sri Lanka | **SN** | Senegal |
| **CN** | China | **LR** | Liberia | **SZ** | Swaziland |
| **CS** | Czechoslovakia | **LT** | Lithuania | **TD** | Chad |
| **CZ** | Czech Republic | **LU** | Luxembourg | **TG** | Togo |
| **DE** | Germany | **LV** | Latvia | **TJ** | Tajikistan |
| **DK** | Denmark | **MC** | Monaco | **TT** | Trinidad and Tobago |
| **EE** | Estonia | **MD** | Republic of Moldova | **UA** | Ukraine |
| **ES** | Spain | **MG** | Madagascar | **UG** | Uganda |
| **FI** | Finland | **ML** | Mali | **US** | United States of America |
| **FR** | France | **MN** | Mongolia | **UZ** | Uzbekistan |
| **GA** | Gabon | **MR** | Mauritania | **VN** | Viet Nam |

# METHOD AND APPARATUS OF REMOVING SUPERIMPOSED LINES AND RESTORING PIXEL VALUES IN IMAGES CONTAINING RECURRING PATTERNS

## FIELD OF THE INVENTION

The present invention relates generally to image processing and more particularly to removing lines superimposed on pixel-array images and, after line removal, restoring pixel values of the image by statistical means. The invention is applicable to images having recurring patterns including particularly textual characters.

## BACKGROUND OF THE INVENTION

Optical character recognition (OCR) and other means to automatically process images find wide application in the processing of labels, forms and other documents containing textual information. An impediment to the application of OCR to the reading of both package labels and forms is the presence of superimposed lines. Frequently the lines are preprinted as underlines on the form and the characters are typed or printed on the form with an incorrect registration, thereby causing the underlines to be superimposed at the base, or even through the middle, of the characters. In other cases, spurious lines can be generated by ink marks, faulty portions of charge belts or laser printers, dirt, and the like. When the superimposed lines pass through or close to characters and other recurring patterns sought to be recognized, OCR algorithms, including those based on neural networks, frequently fail.

Many attempts have been made to solve this problem. A number of known techniques involve simply the removal of the superimposed lines from the image. A standard technique for identifying lines in images is to use algorithms

based on the Hough transform. Usually, detecting a superimposed line in an image is fairly straightforward. However, in the presence of recurring patterns in images, a number of problems can be encountered. In particular, methods of identifying superimposed lines in images containing characters many times offer too many false lines. Attempted resolution of all the false lines dramatically reduces the effectiveness of any method. The adaptive Hough technique, described by J. Illingworth and J. Kittler in "The Adaptive Hough Transform", IEEE PAMI-9, No. 3, pp. 690-698, Sept. 1987, and incorporated herein by reference, attempts to reduce computational requirements by locating superimposed lines at multiple scales of parameter space. However, this and other modifications of Hough-transform methods have not been effective in the presence of characters. Many times the true superimposed line is not even detected in the presence of characters.

Even where line identification and removal is successful, the removal of the lines generally will also remove a portion of the characters or other patterns sought to be recognized. The removal of the character portions frequently generates as many problems as existed before removal of the lines. These problems could be avoided if there were a way to restore the portions of the characters that are removed upon the removal of the superimposed line.

There have been limited attempts to address the restoration of the character portions that are removed upon the removal of a superimposed line. One such attempt is described by J. M. Gloger in "A Hough Transform Based Elimination of Auxiliary Lines in Addresses", United States Postal Service Advance Technology Conference, pp. A-85 - A-92, November 30-December 2, 1992, which is incorporated herein by reference. In this method, line detection and character restoration involve working with the contours of the regions of interest. Lines are detected with the Hough transform. Character restoration is accomplished by connecting the disconnected ends of character contours by the shortest combination of horizontal and vertical line segments. No contextual information along the sequence of characters is used. The present inventor knows of no attempts to use both boundary and context information for character or other pattern restoration.

Howell et al., in U.S. Patent No. 5,226,091, have disclosed the use of finite state machines in general and hidden Markov models in particular to provide a method and apparatus to verify a signature based on position and velocity information obtained during the execution of a number of genuine signatures. However, the '091 patent contains no suggestion or teaching of how finite state

machine techniques might be applied to the problem of restoring missing portions of characters or patterns in images.

Thus a need exists to reliably and efficiently locate superimposed lines in images containing recurring patterns, such as characters, removing the superimposed lines, and replacing the removed lines with image segments that restore the proper shapes to the patterns.

## SUMMARY OF THE INVENTION

The present invention solves the problems of the prior art described above. The invention provides an apparatus and method operating in two stages. In the first stage, superimposed lines in an image comprised of pixels are detected and removed from the image. The removal of the superimposed lines will also remove portions of patterns in the image where the superimposed lines intersected the patterns. In the second stage, the missing parts of the patterns are restored by obtaining the values of valid pixels in a neighborhood of the missing parts of the patterns and performing the restoration by using the values of the valid pixels and a doubly stochastic finite state machine to predict the most likely values for the missing parts of the patterns in the image. The finite state machine is trained prior to use in the restoration with the characters or other patterns sought to be restored.

A finite state machine is a device capable of assuming a finite number of internal states, and which produces an output signal as a function of an input signal and its internal state. A finite state machine is further characterized by the fact that its internal state may change as a result of an input signal. In digital systems, the output signal and the input are typically vectors comprised of a number of discrete, digital values. A digital finite state machine is fully defined by specifying the set of allowable input vectors $I$, the set of allowable output vectors $O$, the set of internal states of the machine $S$, and two functions:

$$\text{Next state function:} \quad I_j \times S_k \rightarrow S_i \quad \text{for all } I_j \in I \text{ and } S_k \in S$$

$$\text{Output function:} \quad I_j \times S_k \rightarrow O_i \quad \text{for all } I_j \in I \text{ and } S_k \in S$$

$$\text{where } S_i \in S \text{ and } O_i \in O$$

In a doubly stochastic finite state machine, the next state and the ouput are not uniquely determined, but occur with given probabilities. That is, for a given input

$I$; and internal state $S_k$, each next state $S_i$ may occur with one certain probability and each output may occur with another certain probability.

In addressing the problem of character or other pattern restoration, two sources of information may be used. The first source is the boundary conditions on either side of the removed line. Any pattern restoration should fit in the observable portion of the patterns. The second source of restoration information is the context of the patterns themselves. Given that the restoration is limited to a sequence of recurring patterns, certain sub-sequences of pattern slices are more likely than others. The use of a finite state machine enables the use of both of these sources of information.

One embodiment of the invention is an apparatus for restoring the missing parts of recurring patterns in an image comprised of pixels, comprising an electronic camera for obtaining the values of valid pixels in a neighborhood of the missing parts of the recurring patterns, a storage device for storing data defining the states of a finite state machine model of the recurring patterns and previously trained with the recurring patterns, and a processor for using the values of valid pixels and at least some of the stored data to restore the values to pixels comprising the missing parts of the recurring patterns.

Another embodiment is a method of line removal and image restoration in an image containing recurring patterns, where the image is comprised of pixels, comprising the steps of detecting lines to be removed at a resolution of one pixel with a generalized Hough transform, locating a cluster of the lines in Hough transform parameter space representing at least one detected line, determining the width and orientation of a thick line corresponding to the cluster, obtaining values of valid pixels in a neighborhood of the thick line, and restoring values to pixels comprising the thick lines using the values of valid pixels and at least a portion of stored data in a finite state machine previously trained with the recurring patterns.

The invention may be used a single stage at a time. Thus one embodiment of the invention is a method of removing lines superimposed on an image comprised of a two-dimensional array of pixels, comprising the steps of detecting the lines at a resolution of one pixel with a generalized Hough transform, locating a cluster of the lines in Hough transform parameter space representing at least one detected line, and determining the width and orientation of a line corresponding to the cluster. Another embodiment of the invention is method of

restoring missing parts of recurring patterns in an image comprised of pixels, comprising the steps of obtaining values of valid pixels in a neighborhood of the missing parts of the recurring patterns, and restoring values to pixels comprising the missing parts of the recurring patterns using the values of valid pixels and at least a portion of stored data in a finite state machine previously trained with the recurring patterns. The recurring patterns may be textual characters. The neighborhood of the missing parts of the recurring patterns can be a finite number of pixels along each of a series of scanning lines passing through the missing parts, the pixels being contiguous or not along each scanning line. The scanning lines may be perpendicular to the superimposed line.

In a preferred embodiment, the invention comprises a charge-coupled device ("CCD") camera to acquire a pixel-array image of a label, form, or other scene containing recurring patterns. The image is stored or transferred to a memory device for processing, preferably by a general-purpose computer. The image may be thresholded so that the pixels in the image take on only one of two values, 0 (for black) and 1 (for white). In a typical thresholded image of an address label, the characters in the address will be made up of black (0) pixels against a white (1) background. The image will contain a number of contiguous black pixels for each character or other pattern to be recognized, with the number of pixels in each pattern depending on the size of the pattern and the resolution of the image.

In the line removal stage of the invention, lines are first detected at the limit of resolution of the image (i.e., one pixel) with a modified version of the Hough transform. The work required to set up the Hough transform accumulator arrays is minimized by estimating image features in a local orientation. The thin lines detected by the Hough transform technique are grouped into clusters in line parameter space. From these line clusters, the parameters of the thick lines such as number of lines to be removed and the width and orientation of each are determined. All of the pixel values making up the thick lines to be removed are marked for subsequent processing.

In a preferred embodiment, the neighborhood of the missing parts of the patterns is provided by scanning along each thick line to be removed. While scanning, a sequence of observation vectors are then mapped into the most probable character restoration using dynamic programming. A Hidden Markov Model ("HMM") comprises the finite state machine that underlies the character restoration. The model is trained from a sequence of character slices somewhat

wider than the thick line. During training, the state sequence is embedded within the observation sequence. Because of the embedding, model probabilities may be estimated by simple relative frequencies.

In this preferred embodiment, two sequences of character slices are used. A training sequence is taken from characters without a superimposed line. Each of the elements of the sequence of training vectors are observable. A run-time sequence is taken about the superimposed line. Elements of the run-time vectors located on the superimposed line are unobservable.

The parameter values for the HMM are estimated from the training sequence. The general technique of training an HMM, Baum-Welch re-estimation, suffers from scaling problems. Because for training the actual states are embedded within the observations, a statistical method can be used. The present invention therefore uses simple relative frequencies to determine both the transition probabilities and the observation probabilities conditioned upon the states. Discussions of these techniques are contained in S. E. Levinson, L. R. Rabiner, M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition", The Bell System Technical Journal, Vol. 62, No. 4, pp. 1035-1074, April, 1983, and L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. IEEE, Vol. 77, No. 2, pp. 257-285, Feb., 1989, which are incorporated herein by reference.

The Viterbi algorithm is employed to determine the most probable sequence of state assignments. At each position slice across the superimposed line, the Viterbi algorithm takes into account two sources of information, the transition probabilities and the observation probabilities conditioned upon the states. Each run-time observation vector could correspond to any number of actual observations. The conditional probability of the run-time observation vector is then the sum of the probabilities of all possible observations. The Viterbi algorithm is discussed in G. D. Forney, Jr., "The Viterbi Algorithm", Proc. IEEE, Vol. 61, No. 3, pp. 268-278, March, 1973, which is incorporated herein by reference.

It is thus an object of this invention to provide a method and an apparatus for quickly and reliably removing superimposed lines from images.

It is a further object of this invention to provide a method and an apparatus for restoring portions of patterns missing from images when superimposed lines are removed from the images.

It is a further object of this invention to provide a method and an apparatus which uses both the boundary conditions on either side of the superimposed line and the context of the patterns themselves to effect image restoration after removal of the superimposed line.

It is a further object of this invention to automatically process images of labels with poor readability due to lines superimposed over characters in the labels so as to generate a high-quality image for OCR input.

The present invention meets these objects and overcomes the drawbacks of the prior art, as will be apparent from the detailed description of the embodiments that follow.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of the major components of a preferred embodiment of the invention.

Figure 2 is a block diagram of the overall operational steps of a preferred embodiment of the pattern restoration portion of the invention.

Figure 3 is a block diagram of the overall operational steps of the line identification and removal portion of a preferred embodiment of the invention.

Figure 4 is a sequence of training observations in an example using a preferred embodiment of the invention.

Figure 5 is a sequence of run-time observations in an example using a preferred embodiment of the invention.

Figure 6 is the organization of finite state machine models for the present invention.

Figure 7 is an example of an input image for the invention.

Figure 8 is an example of restored characters using the invention.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 is a block diagram of the major components of the invention. A camera **11** captures an image of the scene to be processed. The image is stored in an array of pixels, each pixel having a numeric representation of the intensity of light in the portion of the image represented by the pixel. Pixel values vary usually from 0 to 255 or 1023, with the maximum value corresponding to the brightest intensity (white) and with 0 or the minimum value corresponding to the least intensity (black). Pixel arrays may be in any two-dimensional configuration, but typically the most straightforward to use is a rectangular array of rows and columns of pixels.

In the preferred embodiment, camera **11** is a CCD camera, and the output from the CCD is stored in the memory of a general purpose computer **21**. The computer **21** is capable of being programmed to model a finite state machine, the finite state machine being trainable to recognize patterns of the type to be processed by the invention. The computer **21** is further capable of being programmed to carry out the following detailed description of the operation of the invention.

Figure 2 is a block diagram of the overall operational steps of a preferred embodiment of the pattern restoration portion of the invention. Prior to the steps of restoring missing portions of characters in an image, a finite state machine is constructed. As shown in block 1 of figure 2, the finite state machine is capable of representing pixel transitions in neighborhoods of the patterns occurring in the image. Again prior to the restoration steps, the finite state machine is trained with as complete a set as is available of the patterns whose restoration is sought. This step is depicted in block 2 of Figure 2. The construction and training of the finite machine may be accomplished off-line, i.e., independent of the restoration steps and generally only once for a particular pattern set.

The on-line steps of character restoration are depicted on the right side of Figure 2. As shown in blocks 3-7 of Figure 2, pixel data is obtained for an image containing recurring patterns. Additionally, information is obtained regarding which pixels are considered invalid in the image. A pixel may be invalid because its value is missing or because imperfections, such as superimposed lines, are present in the image. Next, valid pixel data is obtained in a neighborhood of one or more invalid pixels. The valid pixel data and the previously trained finite state machine are used to predict values for the invalid pixels, and the predicted values are assigned to the invalid pixels.

Using a Hidden Markov Model ("HMM") as a particular choice of finite state machine, pattern restoration is accomplished by means of two kinds of statistical information, the frequency of occurrence of image pixel value observations on either side of the obscuring line and the frequency of occurrence of pattern slice transitions within the width of the obscuring line. This statistical information is organized as a HMM. Having estimated the model's probabilities, the Viterbi algorithm is then employed to determine the optimal restoration across the length of the line. Prior to the pattern restoration, the HMM is trained. This task involves scanning across sequences of patterns to estimate both observation and transition probabilities. Model construction is done off-line and typically only once.

Also prior to the run-time pattern restoration, the superimposed line itself is located using a modified version of the Hough transform. Figure 3 is a block diagram of the overall operational steps of the line identification and removal portion of a preferred embodiment of the invention. First, lines are detected at the limit of resolution of the image. These thin lines are grouped together in Hough

transform parameter space, identifying the thickness of the detected lines. Local eigenfits in the analysis are employed to reduce the necessary computation.

The line detection and removal operational steps will be discussed in greater detail first, followed by additional detail on pattern restoration for the particular case where the patterns are characters in the English language. The invention works equally well for characters in other languages and for restoring minority missing portions of other recurring patterns such as bar codes, image codes, and the like.

The superimposed lines are located by a Hough transform technique. Hough transforms, reviewed in J. Illingworth, J. Kittler, "A Survey of the Hough Transform", Computer Vision, Graphics, and Image Processing, Vol. 44, pp. 87-116, 1988, which review is incorporated herein by reference, are a known technique in image processing, useful in particular to detect lines in images. In the present invention, lines are detected at the resolution of the image using a Hough transform with a polar parameterization:

$$\begin{aligned} x\cos\Theta + y\sin\Theta &= \rho \\ 0 \le \Theta &\le 2\pi \\ 0 \le \rho &< \infty \end{aligned} \tag{1}$$

For the line detection operation, the image coordinate frame is translated to the center of the image. This allows an unambiguous line parameterization in terms of angle and radius.

Sometimes line orientation may be determined locally. Local orientation estimates are used to reduce the work required to set up the Hough's accumulator array. Given a set of image points with a local coordinate frame located at the center of the point set, the estimation of local orientation may be posed in terms of the minimization of a cost function:

$$E_{\text{local}} = \sum_{x,\,y\ \text{black}} \sum_{x} \sum_{y} (x\alpha + y\beta)^2 - \lambda\left(\alpha^2 + \beta^2 - 1\right) \tag{2}$$

$$\begin{aligned} \alpha &= \cos\Theta \\ \beta &= \sin\Theta \end{aligned} \tag{3}$$

The minimization of $E_{\text{local}}$ is determined by the application of the method of Lagrange multipliers to the least square fit of a line passing through the local origin with a unit length constraint on the line parameters. Define:

$$\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4}$$

$$\vec{p} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{5}$$

Setting the derivative of $E_{local}$ equal to zero implies the necessity to solve an eigensystem. The line orientation, with the orientation specified by the angle of the line's normal with respect to the local origin, is then implied by the eigenvector associated with the smallest eigenvalue.

$$\left( \sum_{\substack{x \\ x, y \text{ black}}} \sum_{y} \vec{x}\vec{x}^{t} \right) \vec{p} = \lambda \vec{p} \tag{6}$$

$$\|\vec{p}\|^2 = 1 \tag{7}$$

Let $\left( \sum_{\substack{x \\ x, y \text{ black}}} \sum_{y} \vec{x}\vec{x}^{t} \right) = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ then the line orientation is:

$$\Theta_{local} = \text{atan}\left(\frac{\beta}{\alpha}\right) = -\text{atan}\left(\frac{a - \lambda_{min}}{b}\right) \tag{8}$$

Since a and c are always greater than or equal to zero.

$$\lambda_{min} = \frac{a + c - \sqrt{(a^2 + c^2 - 2ac + 4b^2)}}{2} \tag{9}$$

To translate the output of the inverse tangent function, usually in the range of $-\pi/2$ to $\pi/2$, into the full range of orientations, 0 to $2\pi$, it is necessary to locate the coordinates of the intersection of the line with the normal from the image center. Performing this step presents no difficulties to those skilled in the art, involving only the solution of a pair of linear equations.

The span of orientation bins of the accumulator array to be updated are proportional to the orientation estimate's uncertainty as indicated by the ratio of eigenvalues:

$$\Theta_{local} - \pi\left(\frac{\lambda_{min}}{\lambda_{max}}\right) \leq \Theta < \Theta_{local} + \pi\left(\frac{\lambda_{min}}{\lambda_{max}}\right) \quad (10)$$

Since the above technique locates lines at the limit of resolution, the located lines are necessarily thin. Where the image contains thick lines instead of thin ones, points in parameter space, each representing a located line, form identifiable clusters. Let each thin line, denoted by a point in parameter space with an accumulation count greater than a threshold, be represented by $\vec{l} = [\rho\ \Theta]^t$. Then an algorithm to identify clusters of thin lines, expressed in pseudo-C code, is:

1. Identify the first line as the first cluster center: $\vec{c}_1 = \vec{l}_1 \cdot C_1 = \{\vec{l}_1\}$

2. For each additional line, $\vec{l}$ {

$$D_j = \min_k \left\| \vec{l} - \vec{l}_k \right\|^2 \qquad \forall C_j, \quad \vec{l} \in C_j$$

If ($\min_j D_j >$ threshold) create a new cluster.

Else update the j[th] cluster where $j$ = argmin $\{D_i\}$ {

$N_j = N_j + 1$, the number of lines in cluster j,

$\vec{c}_j^{new} = \vec{c}_j^{old} + \left(\vec{l} - \vec{c}_j^{old}\right) / N_j$

$C_j^{new} = C_j^{old} \cup \vec{l}$

}

}

The class of model used for character restoration depends upon the line width. For a thick line at any orientation, the line width is determined by:

$$w = \max_{\substack{i,j \\ i \neq j}} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + 1 \quad (11)$$

In the above equation, the $x_i, y_i$ coordinates denote the point of intersection of a cluster member and a line perpendicular to the mean line. Given a point on the mean line, the intersection point with an orthogonal line is found from the solution of a pair of linear equations, well known to those skilled in the art.

A sequence of observation vectors are taken along the mean line. Each observation vector is perpendicular to the mean line. For any point $x_0$ and $y_0$ on the line referred to by the cluster center, the mean line, $[\overline{\rho}\ \overline{\Theta}]^t$, the coordinates of a point offset orthogonally from $x_0$ and $y_0$ by a signed distance $y_1$ are:

$$x = x_0 - y_1 \cos\Theta \tag{12}$$

$$y = y_0 - y_1 \sin\Theta \tag{13}$$

Once the superimposed lines are located, the black pixels representing the line must be set to white in the background regions of the image and left black if within the character portion of the image. Consider a sequence of character slices as a sequence of symbols. Some sort of random process generated the symbol sequence. In the presence of a line through the characters, some portion of the symbols become hidden. Each character slice is an observation. Associated with the hidden portion of each observation is the state of a random process. Character restoration is accomplished by determining the most probable sequence of hidden states.

Character restoration is predicated upon a priori knowledge of the construction of character slice sequences. Such knowledge is embodied in probabilities estimated from character slice sequences taken from line-free regions.

The line-free portions of the image provide the data necessary to train the Hidden Markov Model ("HMM"). Figure 4 is a portion of actual training data. For this case of a five pixel wide line, three additional samples are taken on either side of the line. The samples on either side of the line provide valid data in a neighborhood of the parts of the image that will be missing when the line is removed. Three pixels on either side of the five-pixel-wide line is a preferred number. Potentially more accurate character restoration will result if more than three pixels are used, at a cost of additional storage space and processing time. Conversely, a smaller neighborhood will speed processing but potentially lower the quality of restoration. It is desired that when presented with a line through characters, the restoration algorithm reproduce the most probable pattern sequence within the confines of the line. The training observations are examples of likely pattern sequences with likely states embedded within each pattern.

An HMM defines a doubly stochastic process. In the following discussion, it is assumed that there are N states and M observation symbols. N

equal $2^w$ and M equals $2^m$, where w is the line width and m is the observation symbol width. M is always greater than N. The HMM may be presented by:

$$A = \langle A, B, \pi \rangle \tag{14}$$

A is a N by N matrix of state transition probabilities. The state transition probability is the probability of the next state, given the current state. Each row of A sums to one. $\pi$ is a N dimensional vector of initial state probabilities. The elements of $\pi$ sum to one.

Define $O = \{o_1, o_2, \ldots, o_T\}$ to be a sequence of observation symbols. For training, state examples are embedded within the observations. An extracted state sequence is $Q = \{q_1, q_2, \ldots, q_T\}$.

The goal of training is to maximize the probability of the observation sequence conditioned upon the model parameters. For this problem calculation of simple relative frequencies may substitute for Baum-Welch reestimation, as pointed out by S. Y. Kung, *Digital Neural Networks*, Prentice-Hall, 1993, and which is incorporated herein by reference.

$$A_{i,j} = \frac{\text{count\_of\_transitions\_from\_state\_i\_to\_state\_j}}{\text{total\_number\_of\_transitions\_from\_state\_i}} \tag{15}$$

$$B_{i,j} = \frac{\text{count\_of\_the\_occurrence\_of\_symbol\_j\_when\_in\_state\_i}}{\text{total\_number\_of\_occurrences\_of\_symbols\_when\_in\_state\_i}} \tag{16}$$

$$\pi_i = \begin{cases} 1, & \text{state\_i\_is\_the\_first\_state} \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

For sake of simplicity the simulation assumes that the starting state is always entirely white; i.e., the start state has all ones. Labels usually have a white border making this not an unreasonable assumption.

It is possible that some states and some observation symbols may not appear in the training data. To prevent the failure of the restoration algorithm upon the occurrence of an observation not encountered during training, the *A* and *B* matrices are modified. If a row of *A* is entirely zeros, the state for that row is never observed during training, then all of the transitions from that state are made equally likely.

When in a particular state, an observation symbol can never occur in the state cannot be embedded within the symbol. Hence even without any training data, certain observation symbols have zero probability of occurring from certain states. For those possible observation symbols, a phantom training sequence

precedes the actual training sequence. Those symbols that could occur, but may or may not actually be observed, are given an initial symbol occurrence count of one. The initial occurrence count for impossible symbols is set to zero.

Character restoration after training uses the Viterbi algorithm. The goal of restoration is to find a state sequence that maximizes the probability of the state sequence conditioned upon the observation sequence and the model. Assuming that all observation sequences are equally likely, equivalent results are obtained by maximizing the joint probability of the state sequence and the observation sequence. From the definition of joint probability:

$$P(Q,O|\Lambda) = P(Q|\Lambda) \cdot P(O|Q,\Lambda) \tag{18}$$

In the presence of a line, a portion of run-time observations is shown in Figure 5.

Plugging in the HMM gives the joint probability of a particular state sequence and of a particular observation sequence as:

$$P(Q,O|\Lambda) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t,t+1} \prod_{t=1}^{T} b_{q_t}(o_t) \tag{19}$$

In the above equation, $a_{q_{t,t+1}}$ is the probability of a transition from state $q_t$ to state $q_{t+1}$, an element of the *A* matrix. The line obscures some portion of the observation symbol. Hence any number of complete observation symbols could correspond to the observation vector. The probability of a disjunction of symbols is found by computing the sum of the individual probabilities, $b_{q_t}(o_t)$ is then computed by summing all entries of the *B* matrix on the $q_t$ row that have a symbol matching the observation vector. Symbol positions obscured positions obscured by the line, the 2's of each observation vector, may match either a zero or a one.

Over the length of the line, performing a sequence of additions is more numerically stable than performing a sequence of multiplications. So instead of maximizing the joint probability, it is preferable to minimize the negative logarithm of the joint probability. Any logarithmic base could be used. The natural log is convenient. The following cost function is obtained:

$$E \equiv -\log P(Q,O|\Lambda) = -\left[\log\pi_{q_1} + \log b_{q_1}(o_1) + \sum_{t=1}^{T-1}\left(\log a_{q_t,q_{t+1}} + \log b_{q_{t+1}}(o_{t+1})\right)\right] \quad (20)$$

The obvious way to minimize the cost function is to simply enumerate all possible cases. This implies the necessity to check $2^{wT}$ sequences, which is impractical. A remedy is the utilization of dynamic programming. The Viterbi algorithm uses forward dynamic programming to reduce the number of sequences checked to ($T \times 2^w$), a much more practical amount.

The cost function may be written recursively. Define

$$\delta_{t-1} = -\log P\ (Q,O|\Lambda)|_{\text{up to } t-1}$$

Then the Viterbi algorithm may be summarized as:

$$\delta_1(i) = -(\log\pi_i + \log b_i(o_1))$$

1. Initialization:

$$\Psi_1(i) = 0 \qquad i = 1, 2, \ldots, N$$

$$\delta_t(j) = \min_i\{\delta_{t-1}(i) - \log a_{i,j}\} - \log b_j(o_t) \qquad \begin{matrix} i = 1, 2, \ldots, N \\ j = 1, 2, \ldots, N \end{matrix}$$

2. Forward recursion:

$$\Psi(j) = \arg\min_i\{\delta_{t-1}(j) - \log a_{i,j}\} \qquad t = 2, 3, \ldots, T$$

3. Path backtracking:

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \qquad t = T-1, T-2, \ldots, 1$$

In step three, $q^*$ refers to the optimal state assignment. The sequence of best state assignments as well as the minimum costs are maintained in a doubly linked list called a trellis. (A trellis is discussed in D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, 1987, which is incorporated herein by reference.) Due to the principle of optimality, at each slice of the trellis only the costs of $2^w$ paths need be updated.

Figure 6 shows the method of adjusting the HMM for different thickness lines and different site characters. Lines of an image may be of variable, but uniform width. Line width is determined by the clustering process. The calculated thickness of the line may then direct the restoration algorithm to use the appropriate model. A different training sequence is required for each model.

Generalizing further gives a line of nonuniform width. Such a situation could be handled by having the restoration algorithm switch models upon encountering a change of the line width. No additional training sequences should be necessary. For both levels of generalization the width of the trellis of restored states varies according to the line width.

The HMM for the illustrated restoration is trained on characters having the same size and font as those with the superimposed line. The probabilities of the model can change for characters differing in either their size or their font. The HMM to be used for characters differing in either their size or their font can be generated by trained on the chanced size or shape characters. With sufficient training, it is possible to characterize the dependency of the model probabilities on character size and font type.

Example

A label image containing a superimposed line, shown in Figure 7, was processed by the invention.

The line detection routine used a parameter space threshold of 700 and a cluster threshold of 1.0. Over a 19 by 19 pixel window, eigenfits estimated the local line orientation. Thin lines were clustered into a thick line about 3.4 pixels wide. Two pixels were added to the line width to account for non-uniform digitalization. The line width was then rounded to five pixels. Three pixels on either side of the thick line were included for observations. This gave a total observation symbol size of eleven pixels.

The probabilities of the HMM are estimated from a sequence of over 11,000 training observations. Multiple slices taken across the address label characters are appended together to create the training sequence.

The resulting output is shown in Figure 8. Although there exist a few errors, the restoration appears quite good. Rather noticeable is the complete restoration of the letter E of AVE. Also interesting are the restorations of characters having no examples in the training sequence, such as the letter V and the period. Those characters that remain joined together could probably be separated by recursive OCR dependent segmentation methods. The point of this exercise is to transform the address label into something that the OCR algorithms are more likely to be successful with, while using a minimum of computational effort.

While this invention has been described in detail with particular reference to preferred embodiments thereof, it will be understood that variations and

modifications can be made to these embodiments without departing from the spirit and scope of the invention as described herein and as defined in the appended claims.

5

I claim:

1. A method of restoring missing parts of recurring patterns in an image comprised of pixels, comprising the steps of

obtaining values of valid pixels in a neighborhood of said missing parts of said recurring patterns; and

restoring values to pixels comprising said missing parts of said recurring patterns using said values of valid pixels and at least a portion of stored data in a doubly stochastic finite state machine previously trained with said recurring patterns.

2. The method of claim 1, wherein said recurring patterns are textual characters.

3. The method of claim 1, wherein said neighborhood of said missing parts of said recurring patterns comprises a finite number of pixels along each of a series of scanning lines passing through said missing parts.

4. The method of claim 3, wherein said finite number of pixels are contiguous along each scanning line.

5. The method of claim 1, wherein said missing parts of said recurring patterns are caused by the prior removal of a line superimposed on said image.

6. The method of claim 5, wherein said neighborhood of said missing parts of said recurring patterns comprises a finite number of pixels along each of a series of scanning lines passing through said missing parts.

7. The method of claim 6, wherein said scanning lines are perpendicular to said superimposed line.

8. A method of removing lines superimposed on an image comprised of a two-dimensional array of pixels, comprising the steps of

detecting said lines at a resolution of one pixel with a generalized Hough transform;

locating a cluster of said lines in Hough transform parameter space representing at least one detected line; and

determining the width and orientation of a line corresponding to said cluster.

9. A method of line removal and image restoration in an image containing recurring patterns, said image comprised of pixels, comprising the steps of

detecting lines to be removed at a resolution of one pixel with a generalized Hough transform;

locating a cluster of said lines in Hough transform parameter space representing at least one detected line;

determining the width and orientation of a thick line corresponding to said cluster;

obtaining values of valid pixels in a neighborhood of said thick line; and

restoring values to pixels comprising said thick lines using said values of valid pixels and at least a portion of stored data in a doubly stochastic finite state machine previously trained with said recurring patterns.

10. An apparatus for restoring missing parts of recurring patterns in an image comprised of pixels, comprising

means for obtaining the values of valid pixels in a neighborhood of said missing parts of said recurring patterns;

means for storing data defining the states of a doubly stochastic finite state machine model of said recurring patterns and previously trained with said recurring patterns; and

means, including means for using said values of valid pixels and at least some of the stored data in said means for storing data, for restoring the values to pixels comprising said missing parts of said recurring patterns.

11. An apparatus for removing lines superimposed on an image comprised of a two-dimensional array of pixels, comprising

means for detecting said lines at a resolution of one pixel with a generalized Hough transform;

means for locating a cluster of said lines in Hough transform parameter space representing at least one detected line; and

means for determining the width and orientation of a line corresponding to said cluster.

12. An apparatus for line removal and image restoration in an image containing recurring patterns, said image comprised of pixels, comprising

means for detecting lines to be removed at a resolution of one pixel with a generalized Hough transform;

means for locating a cluster of said lines in Hough transform parameter space representing at least one detected line;

means for determining the width and orientation of a thick line corresponding to said cluster;

means for obtaining values of valid pixels in a neighborhood of said thick line; and

means for restoring values to pixels comprising said thick lines using said values of valid pixels and at least a portion of stored data in a doubly stochastic finite state machine previously trained with said recurring patterns.

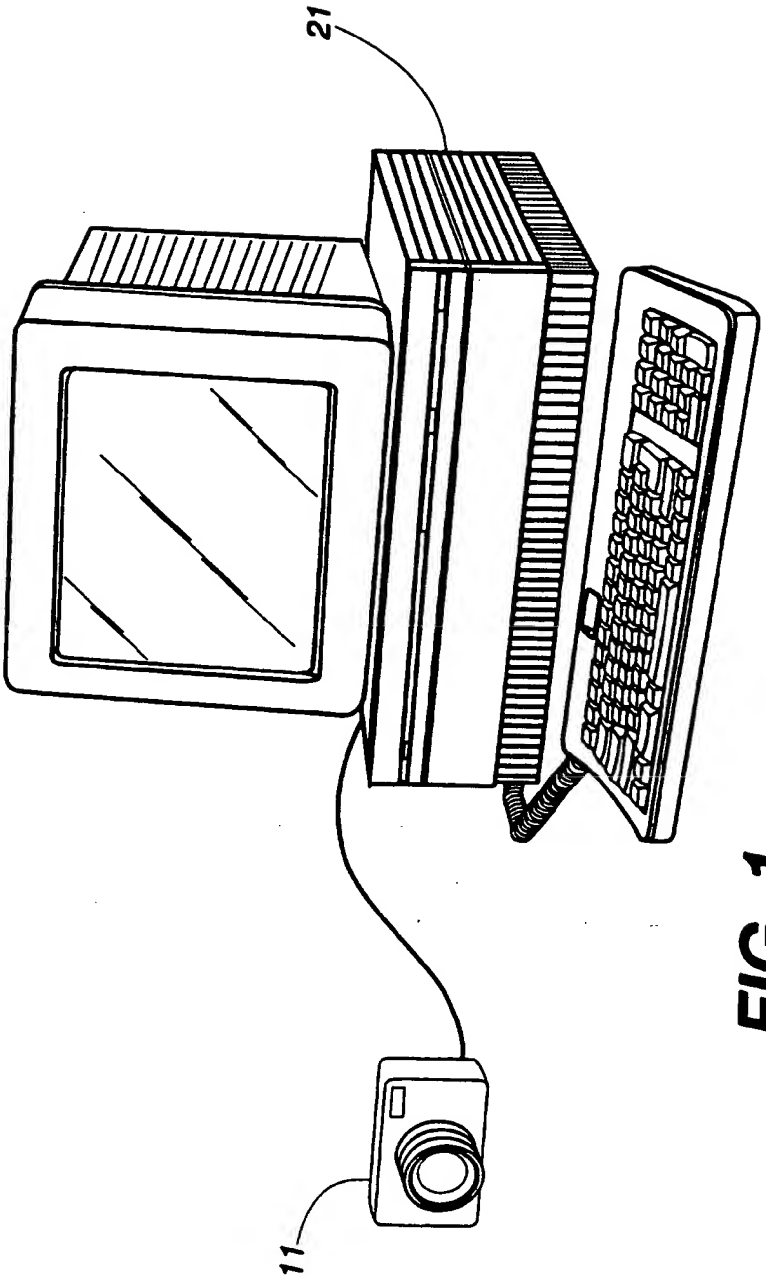



FIG. 1

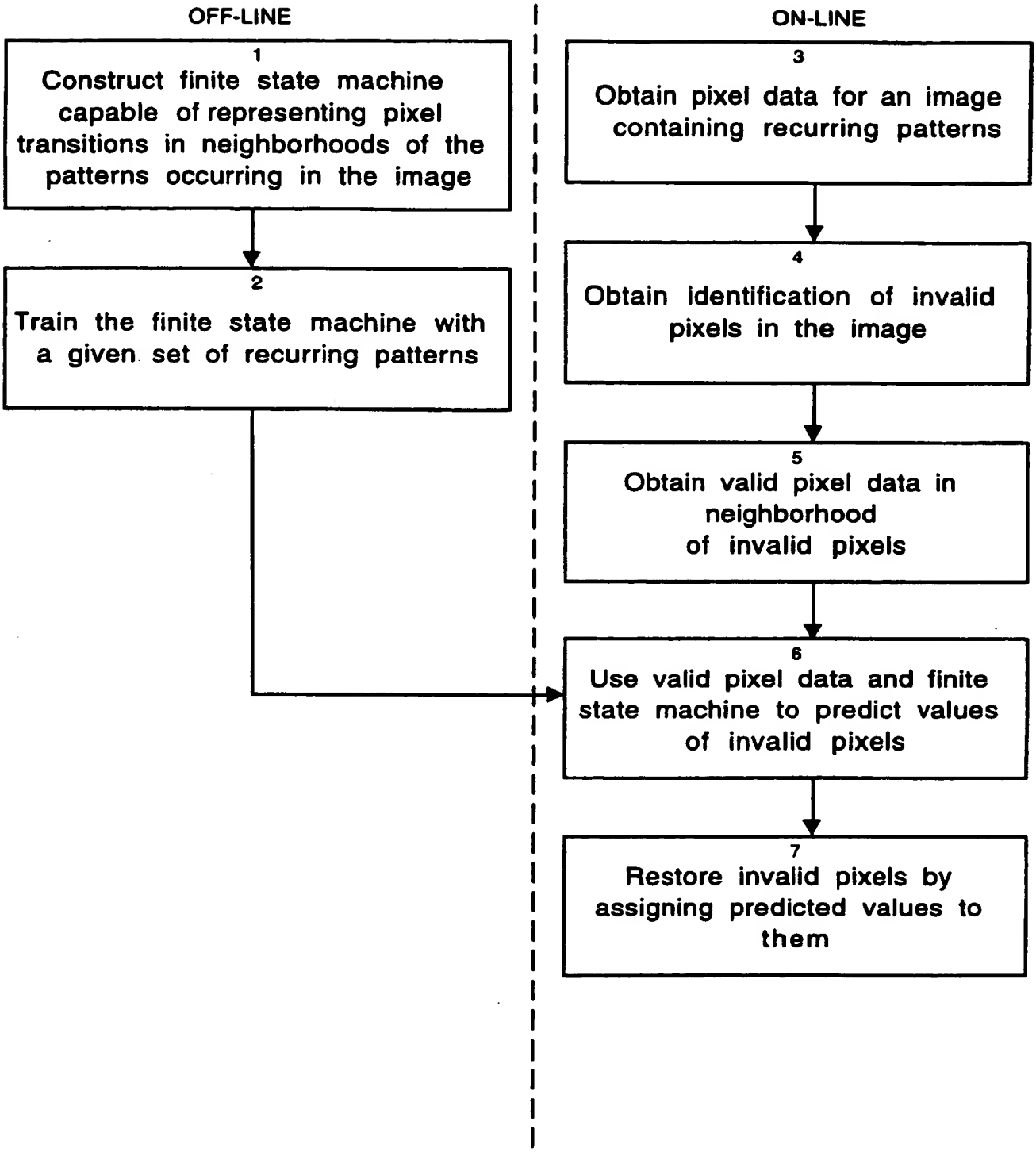

OFF-LINE
ON-LINE
1
Construct finite state machine capable of representing pixel transitions in neighborhoods of the patterns occurring in the image
2
Train the finite state machine with a given set of recurring patterns
3
Obtain pixel data for an image containing recurring patterns
4
Obtain identification of invalid pixels in the image
5
Obtain valid pixel data in neighborhood of invalid pixels
6
Use valid pixel data and finite state machine to predict values of invalid pixels
7
Restore invalid pixels by assigning predicted values to them

FIG. 2

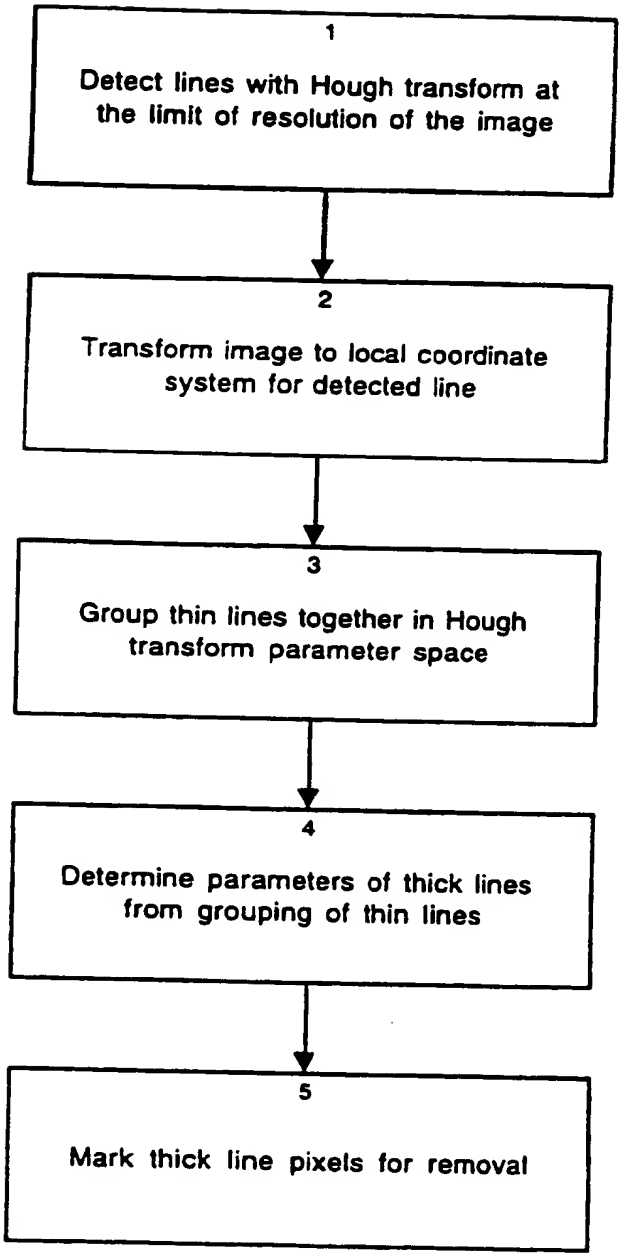

1
Detect lines with Hough transform at the limit of resolution of the image
2
Transform image to local coordinate system for detected line
3
Group thin lines together in Hough transform parameter space
4
Determine parameters of thick lines from grouping of thin lines
5
Mark thick line pixels for removal

FIG. 3

FIG. 4

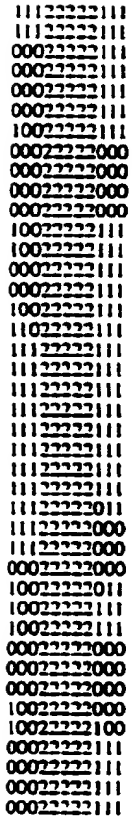


FIG. 5

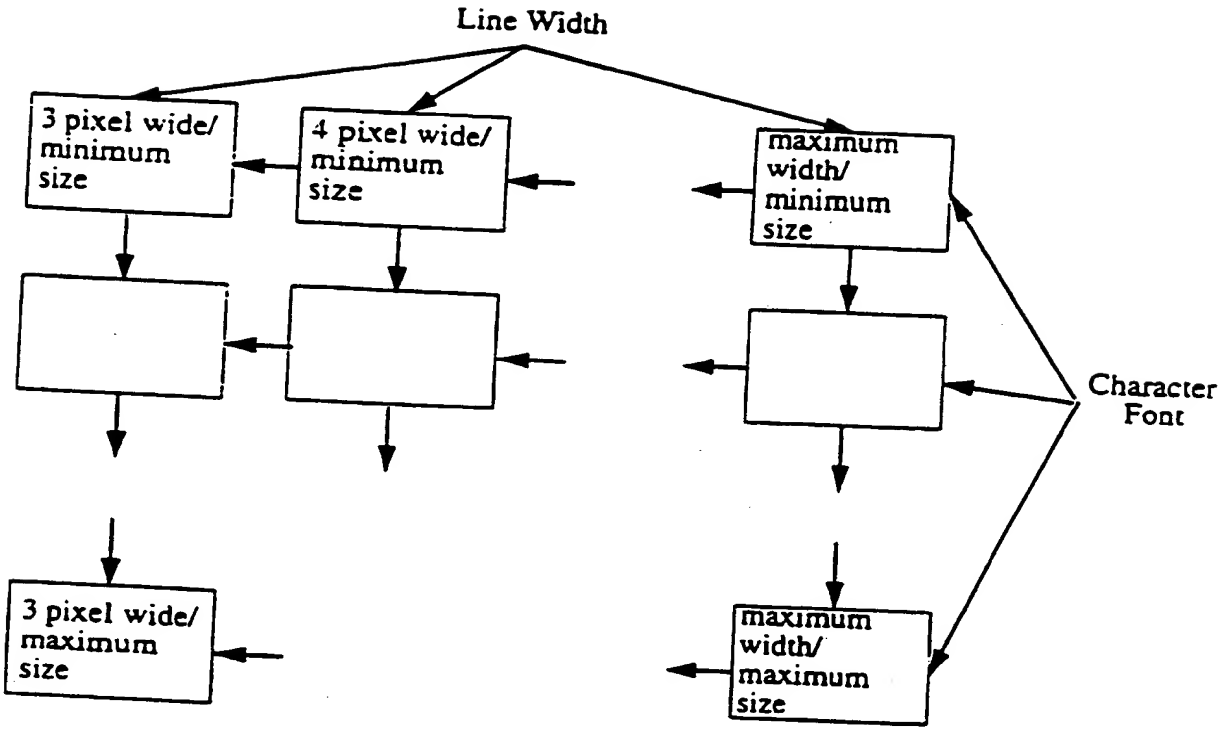

Line Width
3 pixel wide/ minimum size
4 pixel wide/ minimum size
maximum width/ minimum size
Character Font
3 pixel wide/ maximum size
maximum width/ maximum size

FIG. 6

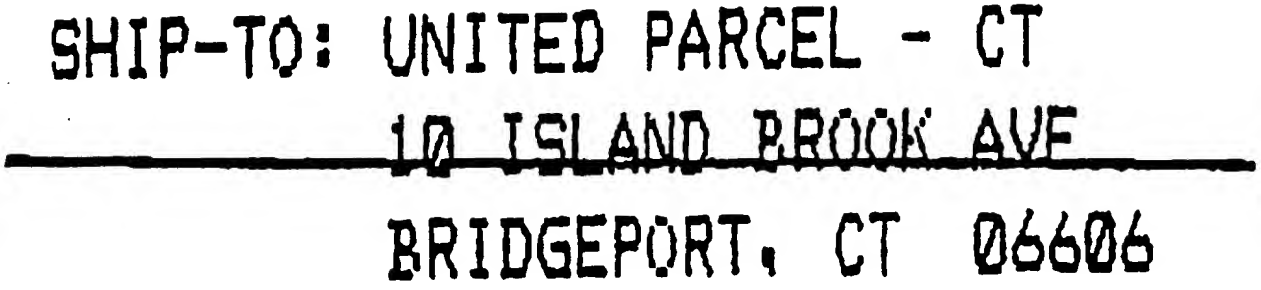

SHIP-TO: UNITED PARCEL - CT
10 ISLAND BROOK AVE
BRIDGEPORT, CT 06606

FIG. 7

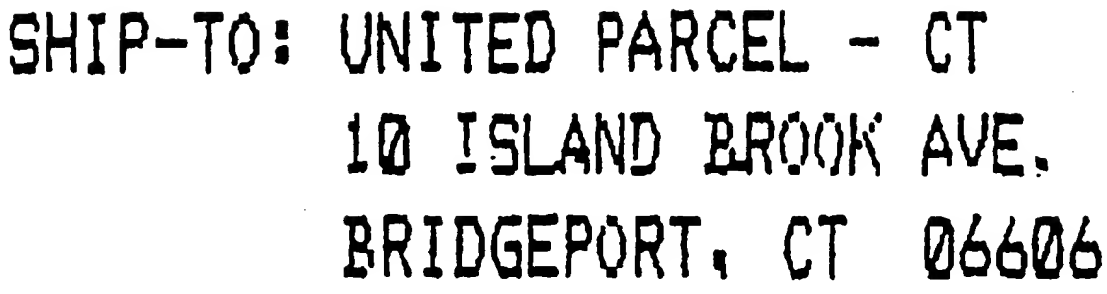

SHIP-TO: UNITED PARCEL - CT
10 ISLAND BROOK AVE.
BRIDGEPORT, CT 06606

FIG. 8